Synthesizing new views from a pair of stereo images

T.-Y. Chao[a], H.-M. Hang[b] and S.-J. Wang[c]

[abc] Dept. of Electronics Engineering,
National Chiao Tung University,
Hsinchu, Taiwan 30050, ROC

ABSTRACT

Our goal in this study is to construct a 3-D model from a pair of (stereo) images and then project this 3-D model to image planes at new locations and orientations. We first compute the disparity map from a pair of stereo images. Although the disparity map may contain defects, we can still calculate the depth map of the entire scene by filling in the missing (occluded) pixels using bilinear interpolation. One step further, we synthesize new stereo images at different camera locations using the 3-D information obtained from the given stereo pair. The disparity map used to generate depth information is one key step in constructing 3-D scenes. Therefore, in this paper we investigate various types of occlusion to help analyzing the disparity map errors and methods that can provide consistent disparity estimates. The edge-directed Modified Dynamic Programming scheme with Adaptive Window (MDPAW), which significant improves the disparity map estimates, is thus proposed. Our preliminary simulations show quite promising results.

Keywords: stereo, 2D/3D conversion, correspondence, disparity estimation, depth estimation, occlusion, dynamic programming, adaptive window, matching indices, 3D modeling

1. INTRODUCTION

Stereo can be a rather broad term; however, stereo, in this paper is used to denote the recovery of a three dimensional scene by using multiple 2D images of the scene. Our purpose is to synthesize new views on different locations and orientations based on this estimated 3-D model.

The overall system structure is first introduced. In the process of estimating disparity map, occlusion is a difficult problem. Hence, we analyze the occlusion and classify the occlusion into two types: limb occlusion and brink occlusion. In addition, these two types of occlusions are explained under two conditions: images obeying the monotonic ordering and images violating the monotonic ordering. Then, we introduce the Cross Correlation (CC) and the Sum of Squared Difference (SSD) matching indices and the dynamic programming (DP) method for image correspondence. Using block matching together with either the CC and SSD matching indices can create smooth disparity map but have poor performance in the smooth image regions. Typically, SSD can generate more smooth disparity map than CC. The DP approach has an excellent performance in the smooth regions, but it produces many occluded pixels scattered over the entire image. When the occluded pixels are interpolated, the high frequency details of the synthesized images are lost. Therefore, we combine SSD and DP to form a method called modified dynamic programming (MDP). The MDP method preserves the advantages of both DP and SSD and avoids some of their disadvantages. One step further, we include an adaptive window in the MDP method, which is called modified dynamic programming with adaptive window (MDPAW). However, the disparity map can be further improved by including edge information in the dynamic programming search process. The simulation results of the edge-directed MDPAW show clear improvement. We use edge-directed MDPAW to estimate the disparity map and the depth information from a pair of (stereo) images and then synthesize the new stereo pairs at new locations and orientations.

2. OVERALL SYSTEM STRUCTURE

Our target is to synthesize new scenes using two given pictures. As shown in Figure 1, the main operation in our system can be partitioned into the following steps. First, a matching technique is used to find the best correspondence between the two given pictures. Second, a disparity map is calculated based on the results obtained in the correspondence step. Then, the next step is to calculate the depth map using 3-D geometric formulas.[1] The forth step is to construct the 3-D object model based on the calculated depth map. Finally, the 3-D objects are projected to the image planes at new locations and orientations.

In addition, because the estimated disparity map contains occluded regions, the missing depth information needs to be filled in using the bilinear interpolation. In synthesizing new view pictures, the unconvered areas have to be filled in by interpolation too. Because the correspondence step (disparity estimate) is very critical — it affects the performance of all the following steps, this paper is mainly focused on solving the correspondence problem. One of our main contributions is to propose better disparity estimation algorithm. Furthermore, the entire system is simulated and many other related issues are discussed in this paper. Although the correspondence problem is rather common in image processing, we found that very few existing literatures discuss the new view synthesizing system. Putting together individual components to form a working system and adjusting them to achieve our goal is our another contribution.

3. OCCLUSION

Occlusion introduces difficulties in performing correspondence. It disturbs the matching process and leads to incorrect disparity estimates, particularly around occluded regions. Basically, we classified occlusions into two major types: limb occlusion and brink occlusion. Then, we discuss in details these two types of occlusions under the monotonic ordering condition. The occlusions violating monotonic ordering are very complex and are discussed only briefly in this paper.

3.1. LIMB OCCLUSION OBEYING MONOTONIC ORDERING

A viewpoint-dependent edge, which is usually called a limb, is an edge with continuous-surface-normal depth discontinuity as shown in Figure 2.[2] We call an occlusion caused by viewpoint-dependent edges a limb occlusion. Assuming the monotonic ordering condition is satisfied, limb occlusion is shown in Figure 3a. We can find that regions I and II in the left image are occluded in the right image. Region I belongs to the background while region II belongs to the foreground. The correct compensation on the reconstructed images for this type of occlusions is to fill in region I using the corresponding background in left image and fill in region II using the corresponding foreground in left image. Hence, to compensate this type of occlusion, we need to find edge A to separate regions I and II. The luminance profile of Figure 3a is shown in Figure 3b. We can find that the boundaries of the occluded part do not occur at the intensity discontinuities. Similar analysis can be performed on regions III and IV. Some previous methods[3] fill region I and II with the background scene. When the occlusion is a limb occlusion and the foreground lacks texture, the pixel matching is likely to be incorrect.

3.2. BRINK OCCLUSION OBEYING MONOTONIC ORDERING

The viewpoint-independent edges are these edges whose three-dimensional spatial locations are independent of the viewpoint. They are usually formed due to the discontinuities on the surface normal, illumination and surface reflectance. The occlusions created by viewpoint-independent edges are called brink occlusion. We can further classify brink occlusions into two classes. In the first class, the viewpoint-independent edges occlude other edges. This case is shown in Figure 4a and its luminace profile is shown in Figure 4b. We find that only one of the two boundaries of the occluded part is not edge points. Hence, the pixel matching in this case should be more accurate than the limb occlusion. The second class is that the viewpoint-independent edge does not occlude any edge except for the background as shown in Figure 5a and Figure 5b. In this class, region I only appears in the left image and it belongs to the background. Therefore, for the second class, compensating region I is easy.

3.3. OCCLUSIONS VIOLATING MONOTONIC ORDERING

The images around occlusions may violate the monotonic ordering. In this case, the limb occlusion and brink occlusion become more complex. In short, in these cases, a nonoccluded area will appear inside the occluded region. If we apply the conventional disparity estimation methods to these cases, the nonoccluded area which violates the monotonic assumption is likely to be treated as a part of the occluded region. Details of the occlusions violating the monotonic ordering are under investigation. For simplicity, we assume the monotonic ordering condition assumption is in this paper.

4. BASIC CORRESPONDING METHOD

Cross Correlation (CC) and Sum of Squared difference (SSD) are two basic matching indices ass defined bellow.

$$SSD(\delta m,\delta n) \equiv \sum\sum_{i,j\in R}[I_l(i,j)-I_r(i-\delta m,j-\delta n)]^2,$$
$$and \quad CC(\delta m,\delta n) \equiv \frac{\sum\sum_{i,j\in R}[I_l(i,j)*I_r(i-\delta m,j-\delta n)]}{\sqrt{\sum\sum_{i,j\in R}I_l^2(i,j)}\sqrt{\sum\sum_{i,j\in R}I_r^2(i-\delta m,j-\delta n)}}. \tag{1}$$

Here, SSD denotes the sum of squared differences over region R with disparity $(\delta m,\delta n)$ and CC denotes the cross-correlation over region R with disparity $(\delta m,\delta n)$. A smaller SSD matching index indicates a better match while a larger CC matching index implies a better match. The matched region is a region in the search range that has the best calculated match. The SSD matching index is more sensitive to the photometric variation and the CC matching index may make more mistakes on regions with similar intensity profiles but different average values (particularly in the smooth region).

The dynamic programming technique is the kernel in this paper. It leads to a considerable computational saving over the exhaustive search in black matching. However, it requires the monotonic-ordering assumption. The dynamic programming method used in stereo image disparity estimation is suggested by Cox, Hingorani and Rao.[4] A brief sketch is described as follows. The matching indices for the dynamic programming method are

$$C_{match} = \frac{(z_1-z_2)^2}{4\delta^2}, \quad C_{occlucion} = ln(\frac{p_d^2\xi}{(1-p_d)\sqrt{2\pi\delta^2}}), \tag{2}$$

where C_{match} is the index for the matched pixel pairs, $C_{occlusion}$ is for the occluded pixels, δ^2 is the variance of the image noise and has a typical value of 4. In eq. (2), z_1 and z_2 are the luminance values captured by cameras 1 and 2, p_d is the probability of the matched pairs (about 0.95), and ξ is the field of view of camera (typically π).
From the above definitions, we can find that the matching cost C_{match} becomes large when (z_1-z_2) is larger or when image noise variance is small. The occlusion cost $C_{occlucion}$ becomes large when the fraction of occlusion pixels is small. The main operation is to calculate $C[i][j]$ (cost function) of two epipolar lines, shown in Figure 6a, by

$$C[i][j] = Min\{(C[i-1][j-1]+C_{match}),(C[i-1][j]+C_{occlusion}),(C[i][j-1]+C_{occlusion}),\} \tag{3}$$

where i represents the horizontal coordinate in the left epipolar line and j represents the horizontal coordinate in the right epipolar line. Hence, $C[i][j]$ (cost function) represents the accumulated matching cost of the first i pixels in the left epipolar line and the first j pixels in the right epipolar line. After $C[i][j]$ (cost function) has been completely calculated for all (i,j), the matching path is reversely traced in the opposite direction from the last calculated pixel pair at the lower right corner and the pixels in the matching path are distinguished into two types: matched pixels and occluded pixels.
For an image epipolar line with N pixels, we perform the following tracing procedure iteratively starting from pixel (N,N).

- **Case 1 :** If $C[i][j] = (C[i-1][j-1] + C_{match})$, then (i, j) is the matching point and the next traced pixel is $(i-1, j-1)$.
- **Case 2 :** If $C[i][j] = (C[i-1][j] + C_{occlusion})$, then the $i'th$ pixel in the left epipolar line is occluded and the next traced pixel is $(i-1, j)$. .
- **Case 3 :** If $C[i][j] = (C[i][j-1] + C_{occlusion})$, then the $j'th$ pixel in the right epipolar line is occluded and the next traced pixel is $(i, j-1)$.

For example, if the current point (M, N) belongs to $Case$ 3, then the next traced pixel is $(M, N-1)$ and we repeat the same operation for $(M, N-1)$ until the entire path is found as shown in Figure 6b. The disparity map generated by using the dynamic programming (DP) method is shown in Figures 10(a). From this disparity map we find that the DP method performs rather well in the smooth region. The most significant evident of DP is that the occluded pixels are scattered over the entire image.

5. IMPROVED DYNAMIC PROGRAMMING METHODS

We modify the existing DP scheme to fit our purpose. The so-called Modified Dynamic Programming Scheme with Adaptive Window (MDPAW) is a combination of the original dynamic programming method and the adaptive matching window technique. The dynamic programming method is pixel-based matching. It can generate a rough disparity map with a small amount of calculations but it often makes mistakes scattered over the entire image. If the bilinear interpolation is used to fill in the occluded pixels, the resultant image will be blurred. To reduce its errors, we use an adaptive window to increase the probability of correct matching. On the one hand, the adaptive window size must be large enough to include sufficient intensity variation for reliable matching. On the other hand, it should be small enough to avoid the effect of projective distortion. In a sense, our MDPAW method is both pixel-based and block-based. However, the MDPAW is not accurate around occlusions. Hence, the edge-directed MDPAW method is developed.

5.1. MODIFIED DYNAMIC PROGRAMMING SCHEME (MDP)

Modified Dynamic Programming Scheme is a combination of the DP method and the block matching method with SSD index. The DP scheme has a high accuracy in matching the smooth regions and repetitive textures but the disparity map itself is not smooth. The SSD index matching method has a poor performance in matching the smooth region but can generate a rather smooth disparity map. Our goal for the MDP scheme is to take the advantages of the DP and the SSD matching methods and to avoid their drawbacks.
The modified matching indices are shown bellow.

$$\begin{aligned} C_{match} &= A \times \frac{1}{BS^2} \sum_{BS} \sum_{BS} \frac{(z_1 - z_2)^2}{4\delta^2}, \\ C_{occlucion} &= ln\left(\frac{(p_d^2 \xi)}{(1-P_d)\sqrt{2\pi\delta^2}}\right), \end{aligned} \tag{4}$$

where BS is the matching block size and the $BS \times BS$ square block is centered at the processing pixel and A is a parameter adjusted empirically. In our simulation, both δ and p_d are not precisely known. Typically, we set $\delta^2 = 4$, $p_d = 0.95$ and $A = 0.04$ for simulation. We find that this disparity map is smoother than the disparity map generated by the DP method and has a better performance in smooth regions than the SSD method. But this disparity map contains errors around image edges. The reason is that the fixed block size is too big for high contrast image regions. The adaptive matching window structure can improve the performance at the edges.

5.2. MODIFIED DYNAMIC PROGRAMMING SCHEME WITH ADAPTIVE WINDOW (MDPAW)

The Modified Dynamic Programming Scheme with Adaptive Window (MDPAW) is a combination of MDP and the adaptive window method. Based on the basic idea of the adaptive windowing, a heuristic window adjusting procedure is proposed bellow. We need to calculate a metric that indicates the local image characteristics. In our scheme, var denotes the variance of the current $BS \times BS$ block in the left image .

- For $var \geq Ta$, $BS = 1$;
- For $var < Ta$ and $var > Tb$, $BS = 3$;
- For $var \leq Tb$, $BS = 5$,

where Ta and Tb are the threshold values. In our experiments, Ta is set to 300 and Tb is set to 50. With the above simple adaptive adaptive window, the value of A in eqn (4) is modified to be

$$A = \frac{BS^2 \times (10 + var)}{5000}. \tag{5}$$

In other words, the A parameter is adjusted according to the image local characteristics. Here, we include var and BS into the A parameter based on our experiences and the performance is quite good. We find that there is significant improvement around the edges. But, incorrect matches still exist (Figure 10b).

5.3. EDGE-DIRECTED MDPAW

In previous discussion of occlusion types, the boundary-matching around occlusions can be classified into edge-to-edge, edge-to-nonedge and nonedge-to-nonedge cases. If we reduce the C_{match} of the matching paths passing through these matching points, then these matching paths have higher probability to survive. Hence, this operation can help increasing the surviving rate of the correct matching path. However, the nonedge-boundaries around the occlusions are difficult to find. We can only reduce the C_{match} of the matching paths passing through the edge-to-edge matching. Based on this idea, we develop the edge-directed MDPAW. From the figures in Section 3, we know that the edge-directed method is more useful in the class II of brink occlusion due to the more frequent edge-to-edge matchings there.

The MDPAW technique can be further improved by taking the image structures into consideration. We include edge information into MDPAW and call this method edge-directed MDPAW. The edge-directed MDPAW method first identifies the edges and then use the edge information to help image matching around the occluded regions. Our approach is to include the edge detection result into the C_{match} metric. Because the occluded regions usually occur around edges, we try to avoid making mismatches in the occluded regions by forcing the corresponding edges to be matched. Hence, we reduce C_{match} when the corresponding points in the left image and the right image are edges. With this modification, the paths passing through the matching edges have higher probability to survive. In this way, the edge information can reduce the probability of mismatch around occluded regions. We define C_{match} to be

$$C_{match} = \frac{1}{BS^2} \sum_{BS} \sum_{BS} \frac{(z_1 - z_2)^2}{4\delta^2} - \frac{edge_l * edge_r * C_{Occlusion}}{BS}, \tag{6}$$

where $edge_l$ equals 1 when the left pixel belongs to an edge and $edge_r$ equals 1 when the right pixel belongs to an edge. The edges are detected by using a gradient filter, whose masks are defined by

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad and \quad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

Method	CC	SSD	DP	MDPAW	edge-directed MDPAW
MSE	505.50	220.58	109.73	62.63	41.68
Reconstructed mean	81.36	82.12	81.76	82.30	82.14
Real mean	82.16				

Table 1. The MSE performance of the test image

Figure 7 explains the basic idea of the edge-directed MDPAW. The horizontal dashed lines represent the position of edges in the left scan lines and the vertical dashed lines represent the position of edges in the right scan lines. The "•" points represent the intersections of the horizontal and vertical dashed lines. Even though these "•" points do not necessarily represent the matched edges, the matching edges usually locate at some "•" points. With this constraint, the paths passing through the matched edges have lower costs because we decrease the C_{match} by $C_{Occlusion}/BS$ there. In addition, when the BS of the matched edge pixel is 1, the $C_{Occlusion}/BS$ will be the largest and has the lowest cost. This happens when pixels locate on real edges. The BS of the pixels away from edges would be larger. To sum up, the correct path has a higher probability to survive. That is, the final path will pass through more matched edges, and mismatched pixels around occluded regions can be decreased. In simulation, Ta is set to 800 and Tb is set to 100 for the edge-directed method.

6. 3D MODELING, 2D PROJECTION, AND SIMULATIONS

Our goal here is to synthesize a pair of new view pictures from two given pictures . As described earlier, our 2D/3D Conversion Scheme is shown in Figure 1. The "left image" and "right image" form a stereo image pair and the "real image" is the image captured by a camera with a rotation angle and camera offset with respect to the left camera shown in Figure 8. We synthesize the "projected image" to approximate the "real image". The "MSE Calculation" unit in this figure is used to evaluate the performance of the entire system. In one special case, we set the rotation angle to be the vergence angle and camera offset to be the baseline. Thus, the "real image" is just the true "right image".

In this figure, the "Model Matching" unit reads in an image pair and generates the corresponding disparity map; the "Depth Calculation" unit calculates the depth map of this image pair and discards the pixels with negative depth values. The "Depth Interpolation" unit inserts the bi-linearly interpolated values into the missing parts of the depth map. Thus, we have the three dimensional information of the entire image. The "Image Projection" unit projects the three-dimensional image onto the two-dimensional plane with the chosen rotation angle and camera offset. In this projection process, some pixels may be missing and the "Interpolation of Projected Image" unit inserts bi-linearly interpolated pixels for the missing pixels. Finally, we evaluate the system performance using the "MSE Calculation" unit if the projected image is available. Due to the lack of well-known good evaluation criterion for the three-dimensional images, we use the 2-D "MSE" to evaluate our 3-D model.

In evaluation, we use the test image pairs shown in Figure 9. The MSE of each method, shown in Table 1, is evaluated on the reconstructed right image based on left image and the estimated 3-D information. We can find that the edged-directed MDPAW method has the smallest MSE. Figure 10 shows the disparity maps generated by using DP, MDPAW and edged-directed MDPAW methods, respectively. We can see that the edged-directed MDPAW method has the best performance around occluded regions. Finally, we show the reconstructed right images generated by using DP, MDPAW and edge-directed MDPAW methods in Figure 11. The image reconstructed by using edge-directed MDPAW retains most image details and the best overall image quality.

7. CONCLUSION

The purpose of this paper is to generate new stereo image pairs from a given set of stereo images. The key step in this process is to estimate the depth information of the given scenes. In order to calculate the depth map, we need

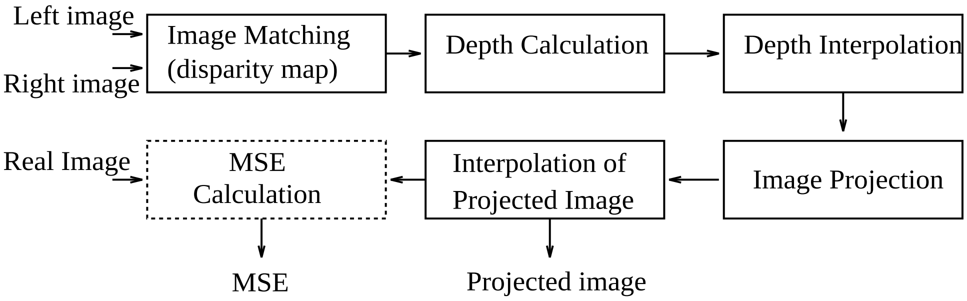


Figure 1. The New View Synthesizing Scheme

to estimate the disparity map first. In this paper, we adopt the dynamic programming (DP) technique in finding the disparity map. However, the direct DP method does not perform well. The MDPAW method is proposed based on a combination of SSD, DP and an adaptive window technique. The SSD method can generate smooth disparity and the DP method is good at smooth region and repetitive texture. The adaptive window approach can improve the performance near the edges without occlusion. One step further, the edge-directed MDPAW includes edge information into MDPAW. From the occlusion analysis, we know that the edge-directed MDPAW method is useful particularly for the class II brink-occlusion. In summary, the edge-directed method has the advantages of the SSD, DP and adaptive window approach, and it also takes into account the edge information to improve the performance near occlusion. Hence, this method has the best performance. However, there is still some room for further improvement in the future.

REFERENCES

1. E. Krotkov, K. Henriksen and R. Kories, "Stereo Ranging with Verging Cameras," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 1200-1205, vol. 12, no. 12, December 1990.
2. V. S. Nalwa, "A Guided Tour of Computer Vision," *Addison-Wesley Publishing Company*, 1987.
3. S. Birchfied and C. Tomasi, "Depth Discontinuities by Pixel-to-Pixel Stereo," *Computer Science Department Stanford Unerversity*
4. I. J. Cox, S. L. Hingorani and S. B. Rao, "A Maximum Likelihood Stereo Algorithm," *Computer Vision and Image Understanding*, Vol. 63, No. 3, pp. 542-567, May, 1996
5. T. Kanade and M. Okutomi, "A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pp. 1088-1094, Sacramento, California, April 1991.
6. T. Kanade and M. Okutomi, "A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 1628-1634, vol. 16, no. 9, Sep. 1994.

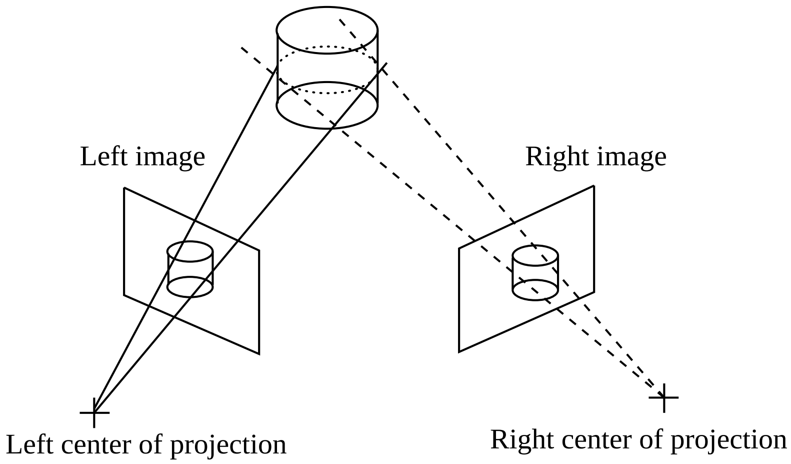


Figure 2. The viewpoint dependent edges

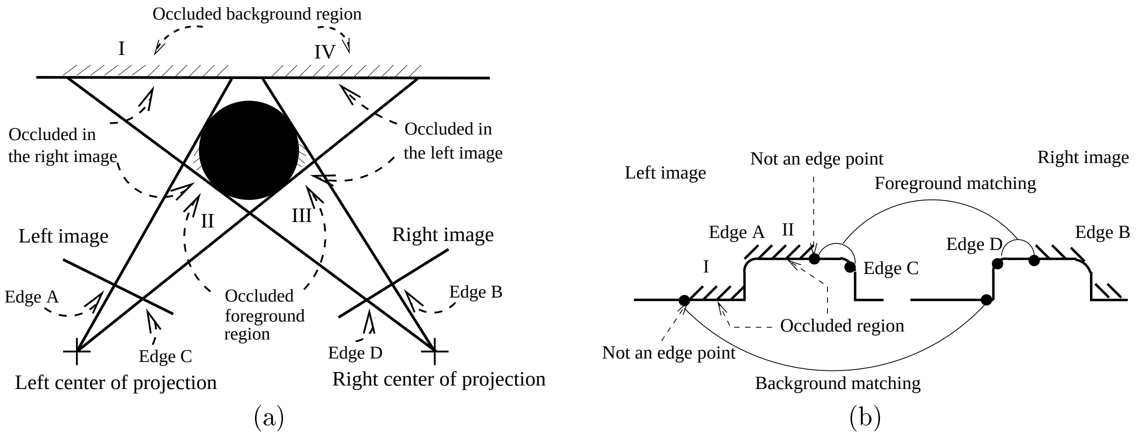


Figure 3. (a) The limb occlusion obeying the monotonic ordering (b) The luminace profile around a limb occlusion obeying the monotonic ordering

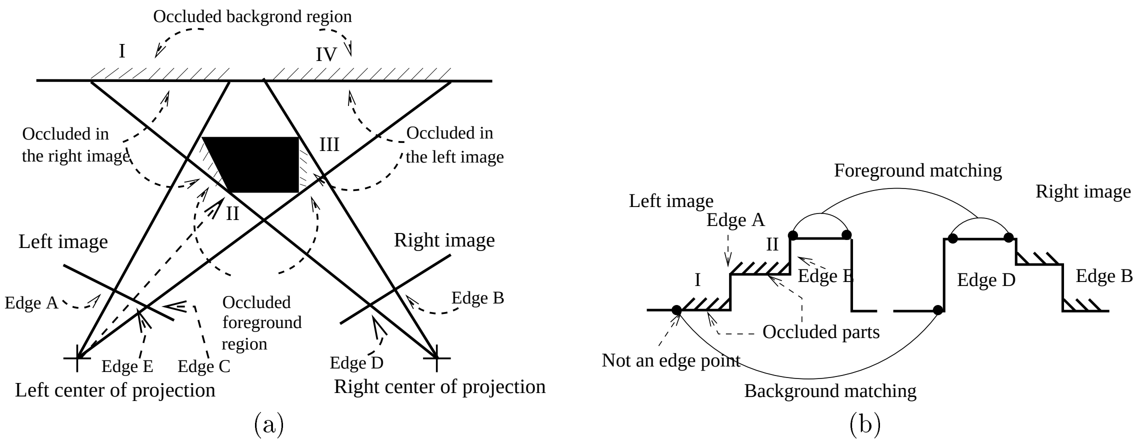


Figure 4. (a) The class I brink-occlusion obeying the monotonic ordering (b) The luminace profile around a brink occlusion the obeying monotonic order: class I

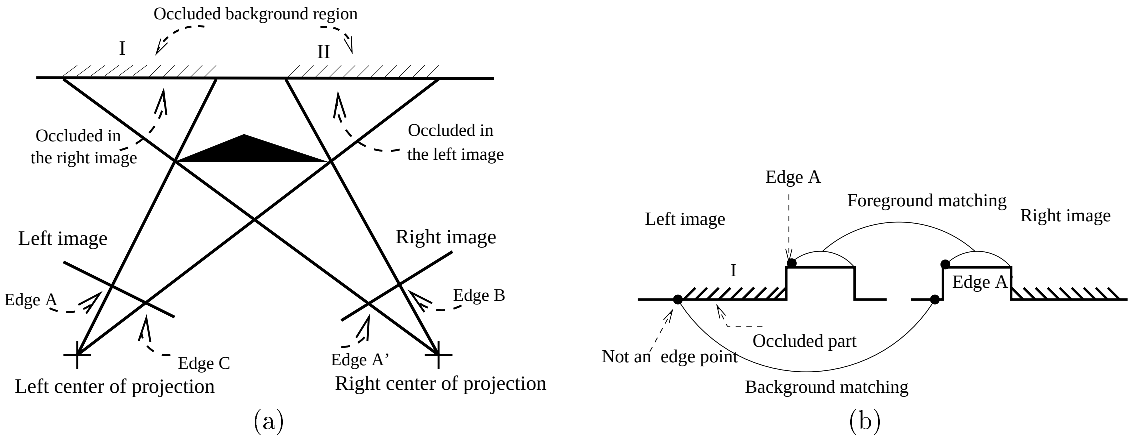


Figure 5. (a) The class II brink-occlusion obeying the monotonic ordering (b) The luminace profile around a brink occlusion obeying the monotonic order: class II

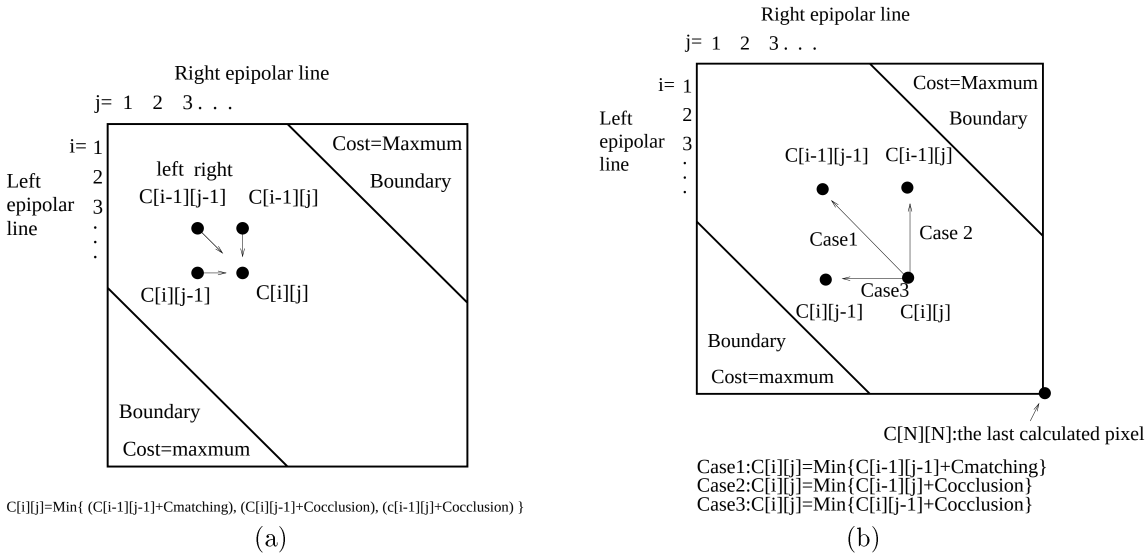


Figure 6. (a) The matching index calculation in dynamic programming (b) The matching path searching in dynamic programming

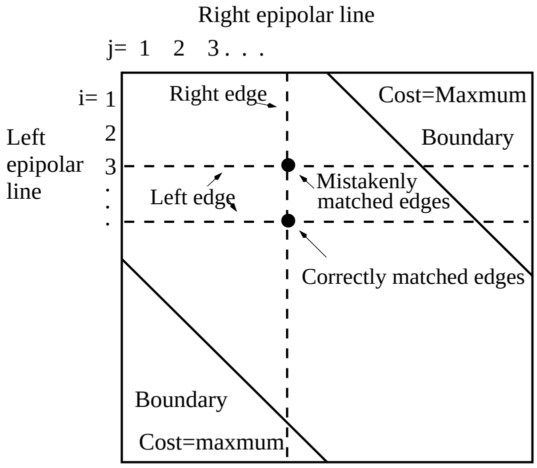


Figure 7. The basic idea of edge-directed MDPAW

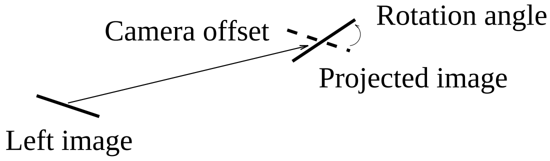


Figure 8. Rotation angle and camera offset

Left

Right

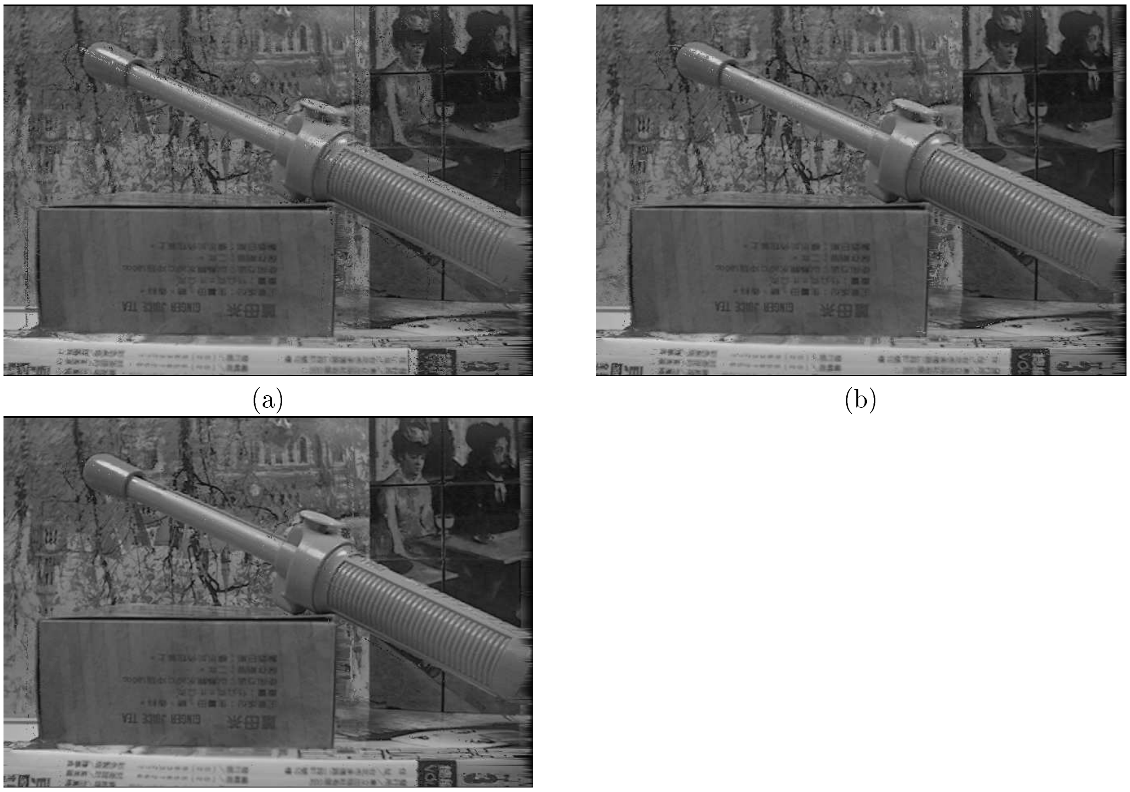
Figure 9. The test image pair

(a)

(b)

(c)

Figure 10. (a)The disparity map generated by DP (b)The disparity map generated by MDPAW (c) The disparity map generated by edge-directed MDPAW



(a) (b) (c)

Figure 11. (a) The reconstructed right image generated by DP (b) The reconstructed right image generated by MDPAW (c) The reconstructed right image generated by edge-directed MDPAW